PBS 用户指南

1、PBS 队列介绍

目前部署在超算集群上有六个队列 ,队列名称分别是 batch、old、oldfat、fat、newfat、gpu。

batch ：默认队列，一般用来做作业测试；

old ：此队列共有 58 个计算节点，每个节点内存为 24G，cpu 核数为 12 核，此队列一般用于对内存需求不太大的作业；
此队列对应的节点为 c0101—c0142 ,c0301—c0332 ,除开有故障的节点，共计 58 个。

oldfat ：此队列共有 1 个胖节点，每个节点内存为 125G，cpu 核数为 32 核；
此队列对应的节点为 c0401。

fat ：此队列共有 3 个胖节点，每个节点内存为 1T，cpu 核数为 40 核，此队列一般用于对内存需求较大的作业；
此队列对应的节点为 fat01、fat02、fat03。

newfat：此队列共有 2 个胖节点，每个节点内存为 1T，cpu 核数为 48 核，此队列一般用于对内存需求较大的作业；
此队列对应的节点为 fat04、fat05。

gpu ：此队列共有 1 个计算节点，每个节点内存为 125G，cpu 核数为 20 核，主要用于视频编码，图形处理等应用。
此队列对应的节点为 gpu01。

2、PBS 命令介绍

PBS 提供 4 条命令用于作业管理。

（1）qsub —— 用于提交作业脚本

命令格式：

```
qsub [-a date_time] [-c interval] [-C directive_prefix]
     [-e path] [-I] [-j join] [-k keep] [-l resource_list] [-m mail_options]
     [-M user_list][-N name] [-o path] [-p priority] [-q destination]
```

[-r c][-S path_list] [-u user_list][-v variable_list] [-V]
[-W additional_attributes] [-z]
[script]

（2）qstat ——用于查询作业状态信息

命令格式：

qstat [-f][-a][-i][-n][-s][-R][-Q][-q][-B][-u]

参数说明：

-f *jobid* 列出指定作业的信息
-a 列出系统所有作业
-i 列出不在运行的作业
-n 列出分配给此作业的结点
-s 列出队列管理员与 scheduler 所提供的建议
-R 列出磁盘预留信息
-Q 操作符是 destination id，指明请求的是队列状态
-q 列出队列状态，并以 alternative 形式显示
-au *userid* 列出指定用户的所有作业
-B 列出 PBS Server 信息
-r 列出所有正在运行的作业
-Qf *queue* 列出指定队列的信息
-u 若操作符为作业号，则列出其状态。若操作符为 destination id，则列出运行在其上的属于 user_list 中用户的作业状态。

常用命令示例：

a) 查看空闲节点信息（提交作业前需要查看一下各队列节点空闲状况）

```
[root@mu01 torque6]# pbsnodes -l free
fat03                 free
fat04                 free
fat05                 free
gpu01                 free
c0106                 free
c0107                 free
c0108                 free
c0109                 free
c0110                 free
c0111                 free
c0112                 free
c0113                 free
c0115                 free
c0116                 free
c0117                 free
```

b) 查看指定作业信息（提交作业后查看作业状态信息）

```
[root@mu01 torque6]# qstat -f 1386
Job Id: 1386.mu01
    Job_Name = work.00035.sh
    Job_Owner = wangwenlab@mu02
    resources_used.cput = 00:00:00
    resources_used.energy_used = 0
    resources_used.mem = 4608kb
    resources_used.vmem = 261920kb
    resources_used.walltime = 16:01:32
    job_state = R
    queue = old
    server = mu01
    Checkpoint = u
    ctime = Mon Apr 16 23:07:20 2018
    Error_Path = mu02:/lustre/home/wangwenlab/yangjie/hudie/fengweidie_project
        /04.correct/homolog/Papilio_machaon.protein.tblastn.shell.22556.qsub/w
        ork.00035.sh.e1386
    exec_host = c0136/0
    Hold_Types = n
    Join_Path = n
    Keep_Files = n
    Mail_Points = a
    mtime = Mon Apr 16 23:07:41 2018
    Output_Path = mu02:/lustre/home/wangwenlab/yangjie/hudie/fengweidie_projec
        t/04.correct/homolog/Papilio_machaon.protein.tblastn.shell.22556.qsub/
        work.00035.sh.o1386
    Priority = 0
    qtime = Mon Apr 16 23:07:20 2018
    Rerunable = True
    Resource_List.feature = HPC
    Resource_List.neednodes = oldPAR
    session_id = 20021
```

c) 查看所有队列作业状态

```
[root@mu01 torque6]# qstat -q

server: mu01

Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
newfat             --      --       --      --    6   0 --   E R
gpu                --      --       --      --    0   0 --   E R
oldfat             --      --       --      --    0   0 --   E R
old                --      --       --      --  113   0 --   E R
batch              --      --       --      --    0   0 --   E R
fat                --      --       --      --    1   0 --   E R
                                                ----- -----
                                                  120     0
```

（3）qdel ——用于删除已提交的作业

命令格式：

qdel [-W 间隔时间] 作业号

命令行参数：

qdel -p 强制清除某个作业号，一般不建议使用

例：# qdel -W 15 211　　15 秒后删除作业号为 211 的作业

（4）qhold & qrls ——作业挂起 & 作业释放

使用 qhold 命令可以挂起作业，使其不被调度执行；

使用 qrls 命令可以将挂起的作业释放，使之可以被调度执行；

命令格式：

qhold *jobid1 jobid2* ...

qrls *jobid1 jobid2* ...

其中 *jobidX* 代表需要操作的作业号，可以一次操作多个作业。

3、PBS 脚本文件

PBS 脚本文件由脚本选项和运行脚本两部分组成。

(1) PBS 作业脚本选项

（若无-C 选项，则每项前面加 '#PBS'）

-a *date_time*　　*date_time* 格式为：[[[[CC]YY]MM]DD]hhmm[.SS]

表示经过 *date_time* 时间后作业才可以运行。

-c *interval*　　定义作业的检查点间隔，如果机器不支持检查点，则忽略此选项。

-C *directive_prefix*　　在脚本文件中以 *directive_prefix* 开头的行解释为 qsub 的命令选项。若无此选项，则默认为 '#PBS'

-e *path*　　将标准错误信息重定向到 *path*

-I　　以交互方式运行

-j *join*　　将标准输出信息与标准错误信息合并到一个文件 *join* 中

-k *keep*　　定义在执行结点上保留标准输出和标准错误信息中的哪个文件。*keep* 为 o 表示保留前者，e 表示后者，oe 或 eo 表示二者都保留，n 表示皆不保留。若忽略此选项，二者都不保留。

-l *resource_list*　　定义资源列表，几个常用的资源种类：

cput=N　　请求 N 秒的 CPU 时间，也可以是 hh:mm:ss 的形式。

mem=N[K|M|G][B|W]　　请求 N {k|m|g}{bytes|words}大小的内存。

nodes=N:ppn=M　　请求 N 个结点，每个结点 M 个处理器。

-m *mail_option*　　*mail_option* 为 a：作业 abort 时给用户发信

为 b：作业开始运行发信

为 e：作业结束运行时发信

若无此选项，**默认为 a**。

-M *user_list*　　定义有关此作业的 mail 发给哪些用户

-N *name*　　作业名，限 15 个字符，首字符为字母，无空格。

-o *path*　　重定向标准输出到 *path*

-p *priority*　　任务优先级，整数，[-1024，1023]，若无定义则为 0

-q *destination*　*destination* 有三种形式：queue；
@server；
queue@server

-r y|n　　指明作业是否可运行，y 为可运行，n 为不可运行。

-S *shell*　　指明执行运行脚本所用的 shell，须包含全路径。

-u *user_list*　　定义作业将在运行结点上以哪个用户名来运行。

-v *variable_list*　定义 export 到本作业的环境变量的扩展列表。

-V　　表明 qsub 命令的所有环境变量都 export 到此作业。

-W *additional_attributes*　　作业的其它属性

-z　　指明 qsub 命令提交作业后，不在终端显示作业号。

(2) 运行脚本同 Linux 下一般的运行脚本文件

[注]：脚本文件中的 mpirun_rsh 命令行中的节点列表文件要用环境变量表示。

$PBS_NODEFILE，这个环境变量表示由 PBS 自动分配给作业的节点列表；节点数为命令行中指定的进程数。

命令格式：

mpirun_rsh -np *进程数* -hostfile $PBS_NODEFILE *可执行程序名*

4、PBS 环境下运行示例

（1）脚本文件编辑示例

实例 1：运行 mpi 程序

命令行：#vi aaa.pbs

编辑的内容：

```
#PBS -N myjob
#PBS -o /home/jz/my.out
#PBS -e /home/jz/my.err
#PBS -l nodes=2:ppn=2
cd 目录（脚本所在的目录）
```

mpirun -np 4 -hostfile $PBS_NODEFILE /home/jz/helloworld

解释：

原来我们都是在终端输入 mpirun；rsh.....这些命令执行程序的，现在只要把这些提交命令放在.pbs 配置文件的最后，由 PBS 来调度执行(自动分配节点和其它资源)。

myjob 是为要运行的程序起的任务名，可以改成你自己想要的名字。原先输出信息都是直接在屏幕上显示的，现在屏幕上的显示全部输出到文件中，上例中输出文件是/home/jz/my.out 文件，可以根据自己的需要修改（目录，文件名）。程序运行时遇到的一些错误会记录在.err 文件中。这样的好处是，因为对每个任务都设定了不同的输出文件，所以看结果只要打开相应文件看就可以了，不需要开多个终端，而且里面有任务的详细信息，比如实际分配的是哪些节点计算，运行时间等。

pbs -l nodes=2:ppn=2

规定使用的节点数，以及每个节点能跑多少核。

mpirun_rsh –np 4 –hostfile $PBS_NODEFILE /home/jz/helloworld

此例中-np 后的 4 是并行数（2×2＝4 个 cpu），-hostfile $PBS_NODEFILE 不需要改变。/home/jz/helloworld 是你编译好的可执行文件名，需修改。

对于每个你要运行的 mpi 程序都需要这样一个.pbs 配置文件，也就是说原来的操作是：mpirun……，现在改成 2 步走：

1）写个 PBS 配置文件（比如 xxx.pbs）；

2）向 PBS 提交（qsub xxx.pbs）

实例 2：运行非 mpi 程序

有些用户并不是自己编写 mpi 程序，同样也可以用 PBS 提交。比如物理系运行程序时一般输入的命令是：

RunDMol3.sh TiFeCp2-pbe-dspp-m=1-opt

那么配置文件可以这样写：

命令行：#vi job.pbs

编辑的内容：

#PBS -N physics_job

#PBS -o /home/physics/physics_job.out

#PBS -e /home/physics/physics_job.err

#PBS –q 队列名称

#PBS -l nodes=1:ppn=2

#PBS -r y

cd 目录（原来直接在节点上运行时所在的目录）
RunDMol3.sh TiFeCp2-pbe-dspp-m=1-opt
解释：

把原来在终端直接输入的命令放到 PBS 配置文件中，因为只要一个节点，所以 nodes=1，至于用哪个节点系统自动分配，可以用 qstat 命令查询（比如 qstat -n）。

用户脚本文件实例：

a)

```
#PBS -N D10
#PBS -o log/D10.out
#PBS -e log/D10.err
#PBS -q old
#PBS -l nodes=5:ppn=4
#PBS -r y
cd /newlustre/home/liaolanjie/yc/12_01_ecard
sh gatk.js D10
```

b)

```
#PBS -N test_GAPE2
#PBS -o /newlustre/home/zhangmao511/workdir/jar_second/jar/my.out
#PBS -e /newlustre/home/zhangmao511/workdir/jar_second/jar/my.err
#PBS -q fat
#PBS -l nodes=1:ppn=10
#PBS -r y
cd /newlustre/home/zhangmao511/workdir/jar_second/jar
java -jar
/newlustre/home/zhangmao511/workdir/jar_second/jar/GAPE_fat_zebraf
ish_Liunx_Beta_V2.jar -c
/newlustre/home/zhangmao511/workdir/jar_second/jar/zebrafish_param
eters.txt
```

（2） 提交作业示例

命令行：#qsub aaa.pbs

（3）作业状态查询示例

qstat 后加不同参数可以查看不同的信息，查看作业的状态。

命令行：#qstat -a

解释：

Job id 211 是给提交的任务分配的任务号，S（常用状态：R 代表运行，Q 代表排队，E 代表正在退出，H 代表挂起，C 代表运行完毕）

命令行：#qstat -n　查看作业使用的节点

命令行：#qstat *jobid1 jobid2* ...　查看指定作业号的作业（可一次查看多个作业）

命令行：#qstat -u *user1*　查看指定用户的作业

解释：该方式输出和默认略有不同，但大同小异。

命令行：#qstat -f *jobid*　查看特定作业详细信息

解释：该命令将会输出作业号为 *jobid* 的作业的详细信息。